

POLY 88 RESIDENT MONITOR VERSION 4.0 11/22/76
COPYRIGHT 1976 INTERACTIVE PRODUCTS CORPORATION
4.0S PAGE 01

```
;
;FINAL VERSION 4.0 MONITOR NOV 22,1976
;
;COPYRIGHT 1976 BY
;POLYMORPHIC SYSTEMS
;A DIVISION OF
;INTERACTIVE PRODUCTS CORPORATION
;737 S. KELLOGG AVE.
;GOLETA, CA 93017
;
;***** 4.0 MONITOR *****
;
;WRITTEN BY          D.L.FAIMAN
;                   D.W.SALLUME
;                   R.L.DERAN
;
;POLY-88 RESIDENT MONITOR ROM VERSION 4.0. RUNS FROM 00
;TO 3FFH IN FIRST CPU ROM SOCKET. FRONT PANEL RESET OR
;THRU ZERO GETS A "POLY-FORMAT" ABSOLUTE TAPE LOADER
;WHICH WILL RUN THE CPU RESIDENT USART TO READ BYTE-STANDARD
;('B') OR "POLY-PHASE" ('P') AUDIO CASSETTE TAPES.
;CONTROL-Z AT ANY TIME ON SYSTEM KBD BRINGS UP FRONT PANEL
;DISPLAY WITH MEMORY MODIFY WINDOW IN HEX. COMMANDS THEN
;ALLOW MEMORY, REGISTER MODIFICATION AND SINGLE-STEP/EXEC
;OF INTERRUPTED PGM.
;
;UTILITY PGMS AVAILABLE FOR USER ARE:
;DSPLY PUTS CHAR ON SYSTEM VIDEO SCR. RECOGNIZES
;      ALL STANDARD CTL CHARS AND SCROLLS TEXT UP.
;      ACCESSIBLE BY CALL TO WORMHOLE 1 (WH1) AT 0C24H.
;KI GETS CHAR FROM SYSTEM CONSOLE KBD.
;      ACCESSIBLE BY CALL TO WORMHOLE ZERO (WH0) AT 0C2
;USRTI GETS CHAR FROM USART BACKGROUND LOAD
;      TYPE ROUTINE.
;HEXO PUTS OUT LOW NIBBLE OF A IN HEX THRU WH1
;BYTE PUTS OUT A IN HEX THRU WH1.
;DEOUT PUTS OUT D,E IN HEX THRU WH1.
;HEXC GETS A HEX NUMBER UP TO 2 BYTES LONG IN H,L FROM
;      WH0, ECHOING ON WH1. NON-HEX CHARACTER
;      TERMINATES INPUT. THE TERM. CHAR IS LEFT IN A.
;MOVE TRANSFERS #BYTES IN BC FROM ADDR IN HL TO ADDR I
;      TERMINATES WITH HL=HL+BC,DE=DE+BC,BC=0.
;IORET A SECTION OF CODE THAT TERMINATES AN
;      INTERRUPT SERVICE ROUTINE BY RESTORING
;      ALL REGS IN STANDARD ORDER, EI, RET.
;TIME EXECUTES 60 TIMES A SECOND AUTOMATICALLY, INCREM
;      ING FOUR BYTE LOCATION CALLED TIMER. IF TIMER=0
;      ALL REGS SAVED AND ROUTINE AT ADDR IN LOCATION
;      WAKEUP IS JUMPED TO. THIS ROUTINE MAY TERMINATE
```

when 4.0 is in RAM,
make chgs on pages

15

HEX016-p8,11,17
chgs

; BY USING IORET.

;*****SYSTEM RAM ALLOCATION*****

;DEDICATED LOCATIONS.

0C00	ORG	0C00H	;BEGIN OF ONBD RAM
0C00	TIMER: DS	4	;INCREMENTED BY 60HZ CLOCK.
			;LOW BYTE=LEAST SIGNIFICANT.
0C04	TANI: DS	4	;TIME AT NEXT INT. USED BY
0C08	TBUFF: DS	2	;USART TRANSMIT BUFFER
			;FLAG AND ONE BYTE BUFFER.
			;USED BY EXTERNAL DUMPERS.
0C0A	RBUFF: DS	2	;USART RECEIVE BUFFER FLAG
			;AND ONE BYTE BUFFER.
			;USED BY MONITOR TAPE LOADER.
0C0C	KBUFF: DS	2	;KBD BUFF-FULL FLAG AND BUFFER
0C0E	POS: DS	2	;CURSOR POSITION FOR DSPLY
			;INTERRUPT SERVICE ROUTINE ADDRESS TABLE.
0C10	SRA1: DS	2	;VI76
0C12	SRA2: DS	2	;VI65
0C14	SRA3: DS	2	;VI54
0C16	SRA4: DS	2	;VI43 USART INT., FIRST INITED SR
0C18	SRA5: DS	2	;VI32 KBD INT.
0C1A	WAKEUP: DS	2	;ADDRESS JUMPED TO WHEN CLOCK TIME
			;OUT IS IN THIS LOC. IT FUNCTIONS
			;EXACTLY LIKE AN SRA, BUT IS ACTUALLY
			;JUMPED TO BY THE CLOCK SOFTWARE.
			;EFFECTIVELY SIMULATES A HARDWARE
0C1C	SRA7: DS	2	;SINGLE-STEP INTERRUPT. NORMALLY
			;AT ITS INITED ADDR WHICH BRINGS
			;UP FRONT PANEL DISPLAY. IT CAN
			;CHANGED FOR EXTRNL PGMS TO USE
			;DEBUGGING OR FANCY PGMING.
			;VIDEO SCREEN ADDRESS PARAMETERS USED IN DSPLY.
			;INITED TO F800H, FCH, BUT MAY BE CHANGED AFTER
			;SYSTEM START UP.
0C1E	SCEND: DS	1	;SCREEN END FOR DSPLY.
0C1F	SCRHM: DS	1	;SCREEN HOME FOR DISPLAY
			;WORMHOLE VECTOR. THIS TABLE IS FULL OF CALL-RETURN PAIR
			;WHICH CALL SYSTEM I/O PROGRAMS. THEY MAY BE CHANGED AFT
			;INITIALIZATION TO CALL ANY PHYSICAL DEVICE DRIVER IT IS
			;DESIRED TO INSTALL. THIS IS NORMALLY DONE WITH LINK PR
			;AS DESCRIBED IN MANUALS. THE FIRST TWO WORMHOLES ARE I
			;TO THE SYSTEM CONSOLE KBD AND SYSTEM VIDEO DISPLAY.
			;ALL USER I/O SHOULD BE DONE THRU THE WH'S TO INSURE
			;COMPATIBILITY WITH ANY SUPERVISOR SYSTEM AND TO ALLOW
			;DYNAMIC REASSIGNMENT OF I/O FOR A FIXED USER OBJECT PGM
0C20	WH0: DS	4	;CONSOLE IN: INITED TO CALL SYS.
0C24	WH1: DS	4	;CONSOLE OUT: INITED TO CALL DSP
0C28	WH2: DS	4	;SYSIN: USED EXTRNL BY TAPE OR
			;INPUT DRIVERS. A CALL GETS BYTE
FFEA	INITLEN EQU	SRA4-\$	;LENGTH OF INITIALIZED MEMORY
0C2C	WH3: DS	4	;SYSOT: USED BY TAPE OR DISK OUT


```

                                ;DRIVERS. A CALL OUTPUTS BYTE IN A
0C30      WH4:      DS      4      ;SYSIN2: SECONDARY SYSTEM INPUT.
0C34      WH5:      DS      4      ;SYSOT2: SECONDARY SYSTEM OUTPUT
0C38      WH6:      DS      4      ;READ: TEXT TYPE INPUT USED
                                ;BY EXTRNL TAPE,DISK,KBD DRIVERS
0C3C      WH7:      DS      4      ;LIST: TEXT TYPE OUTPUT TO LINE
                                ;PRINTERS OR TEXT FILES ETC.
0C40      WH8:      DS      4      ;AUX WH: IN
0C44      WH9:      DS      4      ;AUX WH: OUT
                                ;VCB BLOCKS TO BE TEMPORARILY SWAPPED FOR SCRHM,SCEND AN
                                ;IN ORDER TO USE DSPLY ON OTHER THAN SYSTEM VIDEO DISPLA
                                ;A LINK PGM DOES SWAPPING, AND IS CALLED FROM WH1 IN-
                                ;STEAD OF DSPLY. IT THEN CALLS DSPLY, AND RESTORES
                                ;CONTENTS OF SCRHM,SCEND AND POS BEFORE RET..
                                VCB2:      ;SECONDARY VDO SCRNM CONTEXT
0C48      SCND2:    DS      1
0C49      SCR2:     DS      1
0C4A      POS2:     DS      2
                                VCB3:      ;TERTIARY VDO SCRNM CTX.
0C4C      SCND3:    DS      1
0C4D      SCR3:     DS      1
0C4E      POS3:     DS      2
                                ;
                                ;TEMPORARY LOCATIONS USED BY FRONT PANEL MODE.
                                ;
0C50      WINDOW:   DS      2      ;MEM. MODIFY WINDOW POINTER.
0C52      SAVPC:    DS      2      ;USED BY S.S.WHEN STACK UNAVAILA
                                ;
                                ;TEMPORARY LOCATIONS USED BY TAPE LOADER.
                                ;
0C54      FNAME:    DS      8      ;"FIND" NAME FOR TAPE LOADER.
0C5C      RNAME:    DS      8      ;READ RECD NAME -FOUND ON TAPE.
0C64      RRN:      DS      2      ;READ RECD NUMBER.
0C66      RLEN:     DS      1      ;READ RECD LENGTH.
0C67      RADR:     DS      2      ;READ RECD ADDR (BIAS).
0C69      RTYPE:    DS      1      ;READ RECD TYPE.
                                ;FREE ONBOARD RAM ONWARDS.
                                ;
                                ;***MONITOR PROGRAM***
                                ;
                                ;VECTOR INTERRUPT LOCATIONS
0000      ORG      0
0000 310010  RESET:  LXI      SP,STACK ;FRNT PANEL RST OR POWER UP GET
0003 C30002  JMP      RST1  ;RST1 INITS SYST RAM,ENDS IN TAP
                                ;
                                ;THESE TWO INSTRUCTIONS CAN BE CALLED TO GET THE
                                ;ADDRESS OF THE CALLING PROGRAM INTO H,L. THIS
                                ;IS NECESSARY IN WRITING SELF RELOCATING CODE.
                                ;
0006 E1      POP      H
0007 E9      PCHL
0008 F5      VI6:    PUSH     PSW      ;STANDARD REGISTER PUSH SEQUENCE
0009 C5      PUSH     B
000A D5      PUSH     D

```



```

000B E5          PUSH      H
000C 2A100C      LHLD      SRA1      ;GET SERVICE ROUTINE ADDRESS
000F E9          PCHL
0010 F5          VI5:    PUSH      PSW      ;GO EXECUTE IT. IT WILL RTRN THRU
0011 C5          PUSH      B          ;SAME AS ABOVE      IORET
0012 D5          PUSH      D
0013 E5          PUSH      H
0014 2A120C      LHLD      SRA2
0017 E9          PCHL
0018 F5          VI4:    PUSH      PSW      ;USART
0019 C5          PUSH      B
001A D5          PUSH      D
001B E5          PUSH      H
001C 2A140C      LHLD      SRA3
001F E9          PCHL
0020 F5          VI3:    PUSH      PSW      ;USART
0021 C5          PUSH      B
0022 D5          PUSH      D
0023 E5          PUSH      H
0024 2A160C      LHLD      SRA4
0027 E9          PCHL
0028 F5          VI2:    PUSH      PSW      ;keyBoard
0029 C5          PUSH      B
002A D5          PUSH      D
002B E5          PUSH      H
002C 2A180C      LHLD      SRA5
002F E9          PCHL
0030 F5          VI1:    CLOCK:  PUSH      PSW      ;THE CLOCK INT ALWAYS GOES TO THE
0031 C5          PUSH      B          ;TIMER COUNTER ROUTINE.
0032 D5          PUSH      D
0033 E5          PUSH      H
0034 C34000      JMP      TIME
0037 00          DB      0
0038 F5          SS:    PUSH      PSW
0039 C5          PUSH      B
003A D5          PUSH      D
003B E5          PUSH      H
003C 2A1C0C      LHLD      SRA7
003F E9          PCHL
0040 D308      TIME:    OUT      8          ;ENABLE INT FOR NEXT 60HZ CYCLE
0042 21000C      LXI      H,TIMER ;COUNTER LOCATION
0045 3E04      MVI      A,4          ;4 BYTES TO INCR
0047 34          TIME2:  INR      M          ;INC ONE LOC.
0048 C26400      JNZ      IORET      ;DONE,STANDARD RETURN
004B 23          INX      H
004C 3D          DCR      A
004D C24700      JNZ      TIME2
0050 2A1A0C      LHLD      WAKEUP ;CNTR IS ZERO,SO INITIATE WAKEU
0053 E9          PCHL          ;GO TO THE WAKEUP TASK
          ;USART INTERRUPT SERVICE ROUTINE FOR INPUT
          ;ITS ADDRESS IS INITED INTO SRA4.
0054 DB01      USRTSR:  IN      1
0056 E602      ANI      2
0058 CA6400      JZ      IORET

```


005B DB00 IN 0
 005D 210A0C LXI H,RBUFF

; THIS CODE IS SHARED BY ANY SERVICE ROUTINE
 ; WHICH HAS FOUND A CHARACTER. HL SHOULD HAVE THE
 ; BUFFER ADDRESS, FLAG FIRST-THEN ONE BYTE BUFF.
 ; ONLY USED BY INPUT ISR'S.

0060 3600 IOPUT: MVI M,0 ; ZERO THE FLAG: WE GOT THE CHAR
 0062 23 INX H ; MOVE UP TO DATA BUFFER
 0063 77 MOV M,A ; PUT CHAR IN DATA BUFF
 ; FALL THRU TO IORET
 0064 E1 IORET: POP H ; THIS CODE SECTION CAN BE JUMPED
 ; FOR STANDARD SEQ.REG.POPS AND E
 ; AS FOR ANY INTERRUPT SERVICE RT
 0065 D1 POP D
 0066 C1 POP B
 0067 F1 POP PSW
 0068 FB EI
 0069 C9 RET

; WORMHOLE END OF KBD CHAR FETCH RTNS.
 ; WHEN CALLED BY WH0, WAITS FOR VALID DATA FLG IN KBUFF,
 ; THEN RETURNS CHAR FROM KBUFF+1 IN A.

006A E5 KI: PUSH H ; WE CAN ONLY WRECK A, SO SAVE H
 006B 210C0C LXI H,KBUFF
 006E 7E KIL: MOV A,M ; GET FLAG
 006F B7 ORA A ; IS IT ZERO?
 0070 C26E00 JNZ KIL ; NO, TRY AGAIN
 0073 35 DCR M
 0074 23 INX H
 0075 7E MOV A,M
 0076 E1 POP H
 0077 C9 RET
 0078 E5 USRTI: PUSH H
 0079 210A0C LXI H,RBUFF
 007C C36E00 JMP KIL

; ***** VIDEO DISPLAY DRIVER *****

; DSPLY IS THE FAMOUS "TELETYPE SIMULATOR" WHICH
 ; DRIVES A POLYMORPHICS VTI AND SCROLLS TEXT WHEN
 ; THE SCRNM FILLS. IT RECOGNIZES ALL IMPORTANT CONTROL
 ; CHARACTERS AND USES THE SCRHM, SCEND, AND POS LOCATIONS
 ; IN SYSTEM RAM WHICH ARE INITIALIZED TO GIVE A SCREEN
 ; AT 0F800H TO 0FBFFH ON POWER-UP OR FPRST. AT
 ; THE SAME TIME, ITS OWN ADDRESS IS PUT IN WH1 FOR THE
 ; DEFAULT SYSTEM CONSOLE DISPLAY DRIVER.

007F F5 DSPLY: PUSH PSW ; A WH PGM MUST SAVE ALL REGS
 0080 C5 PUSH B
 0081 D5 PUSH D
 0082 E5 PUSH H
 0083 2A0E0C LHLD POS
 0018 CTLX EQU 18H ; ASCII "CAN" CLEARS CUR. LINE

0086 FE18		CPI	CTLX	
0088 CAD000		JZ	CLINE	
008B 11B900		LXI	D, SCRL	; ALL DSPLY CHARACTER
008E D5		PUSH	D	; HANDLING SUBROUTINES EXCEPT
				; CLINE RETURN TO SCRL TO CHECK
				; SCRN OVERFLOW AND SCROLL
				; IF NECESSARY.
008F 367F		MVI	M, 07FH	; BLANK THE CURSOR SO WE
				; WON'T HAVE TO LATER.
				; USE A GRAPHICS BLANK
				; INSTEAD OF A SPACE.
007F	RB	EQU	7FH	; ASCII "DEL" OR RUBOUT
				; BACKSPACE AND DELETE CHAR.
0091 FE7F		CPI	RB	
0093 CAE400		JZ	RBR	
0096 FE20		CPI	20H	; IF CHAR IS ABOVE 20H, PRINT IT
0098 D2B400		JNC	NORM	; ELSE IT'S NOT A NORMAL CHAR,
				; SO TEST IT FOR VALID CTL CODE
000D	CR	EQU	0DH	; CARRIAGE RETURN. MOVES DOWN
				; A LINE, LEFT-ZEROES CURSOR.
009B D60D		SUI	CR	
009D CAE600		JZ	CRR	
000C	FF	EQU	0CH	; FORM FEED. CLEARS SCRN, HOMES
				; CURSOR.
00A0 3C		INR	A	
00A1 CAEF00		JZ	FFR	
000B	VT	EQU	0BH	; VERTICAL TAB. JUST HOMES CURSOR
00A4 3C		INR	A	
00A5 CAF900		JZ	VTR	
0009	TAB	EQU	09H	; TAB. MOVES CURSOR RIGHT TO
				; NEXT EVEN/8 POSITION.
00A8 3C		INR	A	
00A9 3C		INR	A	
00AA C0		RNZ		; IF NOT A TAB, RETURN
00AB 7D		MOV	A, L	; BACK UP CURSOR TO EVEN/8
00AC E6F8		ANI	0F8H	
00AE 6F		MOV	L, A	
00AF 010800		LXI	B, 8	; MOV UP 8 POSITIONS
00B2 09		DAD	B	
00B3 C9		RET		
00B4 F680	NORM:	ORI	80H	; WE HAVE A PRINTING CHAR, SO MAP
				; IT INTO CHAR AREA OF VTI SPACE.
00B6 77		MOV	M, A	; PUT IT IN REFRESH MEM.
00B7 23		INX	H	; MOVE UP A POSITION.
00B8 C9		RET		; GO TO SCRL.
				; SCRL SCROLLS TEXT UP THE SCRN IF NECESSARY,
				; THEN FALLS INTO CURP, WHICH RESTORES THE CURSOR
				; AND RETURNS TO USER THRU IORET.
				; SCRL:
00B9 3A1E0C		LDA	SCEND	
00BC BC		CMP	H	
00BD C2DC00		JNZ	CURP	
00C0 2A1E0C		LHLD	SCEND	


```

00C3 7C      MOV      A,H
00C4 95      SUB      L
00C5 54      MOV      D,H
00C6 2E40    MVI      L,40H
00C8 1E00    MVI      E,0
00CA 4D      MOV      C,L
00CB 47      MOV      B,A
00CC CD0001  CALL     MOVE
00CF 2B      DCX      H

;
;CLINE CLEARS THE CURRENT LINE
;
00D0 3E3F    CLINE:  MVI      A,3FH
00D2 57      MOV      D,A
00D3 B5      ORA      L
00D4 6F      MOV      L,A
00D5 367F    WIPE:   MVI      M,7FH
00D7 2B      DCX      H
00D8 15      DCR      D
00D9 C2D500  JNZ      WIPE
00DC 36FF    CURP:   MVI      M,0FFH
00DE 220E0C  SHLD     POS
00E1 C36400  JMP      IORET
00E4 2B      RBR:    DCX      H ;RUBOUT ROUTINE
00E5 C9      RET
00E6 014000  CRR:    LXI      B,64 ;CARRIAGE RETURN RTN.
00E9 7D      MOV      A,L
00EA E6C0    ANI      0C0H
00EC 6F      MOV      L,A
00ED 09      DAD      B
00EE C9      RET
00EF CDF900  FFR:    CALL     VTR ;FORM FEED ROUTINE
00F2 367F    FF1:   MVI      M,7FH
00F4 23      INX      H
00F5 BC      CMP      H
00F6 C2F200  JNZ      FF1
00F9 2A1E0C  VTR:    LHLD     SCEND ;VERTICAL TAB RTN.
00FC 7D      MOV      A,L
00FD 2E00    MVI      L,0
00FF C9      RET

;
;MOVE MOVES -BC BYTES FROM THE AREA STARTING AT
;(HL) TO THE AREA STARTING AT (DE)
;
0100 7E      MOVE:   MOV      A,M
0101 12      STAX     D
0102 13      INX      D
0103 23      INX      H
0104 0C      INR      C
0105 C20001  JNZ      MOVE
0108 04      INR      B
0109 C20001  JNZ      MOVE
010C C9      RET

```


;KEYBOARD INTERRUPT SERVICE ROUTINE FOR SYSTEM CONSOLE
 ;KEYBOARD AT 0F8H. ITS ADDRESS IS INITED INTO SRA5.
 ;IT WATCHES FOR CTL/Z. IF IT FINDS ONE, IT FALLS
 ;THRU INTO SAVE, THUS ENTERING FRONT PANEL MODE.
 ;KBD ADDR IS FIXED BY SCREEN ADDR, WHICH IS 0F800H
 ;SINCE A POLYMORPHIC VTI USES SAME DECODER FOR KBD
 ;AS FOR VIDEO REFRESH MEMORY.

010D DBF8
 010F 210C0C
 001A
 0112 FE1A
 0114 C26000

KSR: IN 0F8H
 LXI H,KBUFF
 CTLZ EQU 01AH
 CPI CTLZ
 JNZ IOPUT

;***** FRONT PANEL MODE *****

;SAVE IS THE ENTRY POINT INTO FRONT PANEL
 ;MODE IF REGISTERS HAVE BEEN ALREADY PUSHED IN
 ;STANDARD SEQUENCE. IT PUSHES PC AND SP
 ;ON TOP OF REGISTERS FOR DISP TO USE.

0117 210A00
 011A 39
 011B E5
 011C 2B
 011D 56
 011E 2B
 011F 5E
 0120 D5

SAVE: LXI H,10
 DAD SP
 PUSH H
 DCX H
 MOV D,M
 DCX H
 MOV E,M
 PUSH D

;WARM IS THE ENTRY POINT TO FRONT PANEL MODE
 ;IF PC AND SP HAVE ALREADY BEEN PUSHED DOWN
 ;ON TOP OF REGISTERS IN STANDARD SEQUENCE

0121 CD9203
 0124 FB
 0125 CDF900
 0128 015101
 012B 09
 012C 369C
 012E 017501
 0131 09
 0132 369A

WARM: CALL CLEAR
 EI
 CALL VTR ;SET HL TO BEGINNING OF SCREEN
 LXI B,337 ;OFFSET FROM SCRHM FOR UPARROW
 DAD B
 MVI M,9CH ;VTI CODE FOR UPARROW
 LXI B,175H ;OFFSET FROM UPARROW FOR
 DAD B ;RIGHT ARROW
 MVI M,9AH ;VTI CODE FOR RIGHT ARROW

;DISP IS THE ENTRY POINT TO FRONT PANEL MODE IF
 ;THE SCREEN ALREADY SHOWS A FRONT PANEL DISPLAY.
 ;IT DOES NOT CLEAR THE SCREEN, SO WILL NOT
 ;BLINK WHEN SCREEN IS UPDATED.

0134 3E18
 0136 CD7F00
 0139 3E0B
 013B CD7F00

DISP: MVI A,CTLX ;ERASE LAST COMMAND ON SCRNM
 CALL DSPLY
 MVI A,0BH
 CALL DSPLY

013E 3E06	MVI	A,6
0140 210000	LXI	H,0
0143 39	DAD	SP
0144 016D01	LXI	B,MSG
0147 F5	PUSH	PSW
0148 37	DISPl: STC	
0149 CD7C01	CALL	FLDSY
014C CD7C01	CALL	FLDSY
014F CD9703	CALL	BLK
0152 5E	MOV	E,M
0153 23	INX	H
0154 56	MOV	D,M
0155 23	INX	H
0156 CDD103	CALL	DEOUT
0159 F1	POP	PSW
015A 3D	DCR	A
015B CA8701	JZ	FLAGS
015E F5	PUSH	PSW
015F C5	PUSH	B
0160 EB	XCHG	
0161 01FDFE	LXI	B,-3
0164 09	DAD	B
0165 EB	XCHG	
0166 CD7F03	CALL	HEX08
0169 C1	POP	B
016A C34801	JMP	DISPl
016D 50435350 MSG:	DB	'PCSPHLDEBCAFCMZ'
0171 484C4445		
0175 42434146		
0179 434D5A		

;FIELD DISPLAY PUTS OUT CHAR IN ADDR IN B IFF CY SET.
 ;IFF CY ZERO, PUT BLANK TO VDO DISPLAY.
 ;IN EITHER CASE, B IS INCREMENTED.

017C F5	FLDSY: PUSH	PSW
017D D49703	CNC	BLK
0180 0A	LDAX	B
0181 DC7F00	CC	DSPLY
0184 F1	POP	PSW
0185 03	INX	B
0186 C9	RET	
0187 CD9C03	FLAGS: CALL	TABBER
018A 7B	MOV	A,E
018B 0F	RRC	
018C CD7C01	CALL	FLDSY
018F 07	RLC	
0190 07	RLC	
0191 CD7C01	CALL	FLDSY
0194 07	RLC	
0195 CD7C01	CALL	FLDSY

;MMOD PLACES THE MEMORY MODIFY DISPLAY
 ;ON THE SCREEN WINDOW POINT TO THE BYTE

end of top half display

;TO BE MODIFIED

```

;
MMOD:  CALL    CROUT
      CALL    CROUT
      LHLD    WINDOW
      LXI     B,-32
      DAD     B
      MVI     C,8
      XCHG
MMOD1: CALL    DEOUT
      CALL    TABBER
      MVI     A,7FH
      CALL    DSPLY
      CALL    HEXO8
      DCR     C
      JNZ     MMOD1
  
```

;COMD CALLS HEXC TO GET HEX NUMBERS TO BE ENTERED INTO M
 ;THEN EVALUATES THE NON-HEX TERMINATING CHARACTER TO SEE
 ;IT IS A COMMAND

```

;
COMD:  CALL    HEXC
      MOV     A,C
      ORA     A
      CNZ     STORE
      MOV     A,B
      LXI     H,DISP
      PUSH    H
      LXI     H,MTBL
      CALL    LOOKUP
      MVI     D,0
      JNC     ADDIT
      CALL    LOOKUP
      RC
      LXI     H,RTN
      DAD     D
      PCHL
  
```

;LOOKUP COMPARES THE ACCUMULATOR
 ;AGAINST THE ENTRY IN A TABLE
 ;POINTED TO BY HL AND RETURNS WITH
 ;THE BYTE FOLLOWING IT IN E.
 ;CARRY FLAG SET IF NO MATCH
 ;TABLE MUST END IN 0FFH, EACH ENTRY IS 2 BYTES

```

;
LOOKUP: CMP     M
      INX     H
      MOV     E,M
      RZ
      INX     H
      MOV     E,M
      INR     E
      JNZ     LOOKUP
      INX     H
  
```


01E6 37
 01E7 C9

STC
 RET

; MTBL IS THE TABLE OF COMMANDS FOR
 ; MEMORY MODIFY

01E8 20	MTBL:	DB	' '	; SPACE MOVE POINTER FORWARD
01E9 11		DB	17	
01EA 08		DB	8	; BACKSPACE MOVES BACKWARDS ONE
01EB 0F		DB	15	
01EC 0D		DB	13	; RETURN MOVES FWD 8 BYTES
01ED 18	20 FOR HEXD16	DB	24	
01EE 0A		DB	10	; LINE FEED MOVES BACK 8 BYTES
01EF 08	0 FOR HEXD16	DB	8	
01F0 FF		DB	-1	; END OF 1ST HALF TABLE
01F1 47		DB	'G'	; GO
01F2 6C		DB	G-RTN	
01F3 53		DB	'S'	; SET REGISTER
01F4 7C		DB	SETR-RTN	
01F5 58		DB	'X'	; EXECUTE (SINGLE STEP)
01F6 59		DB	X-RTN	
01F7 49		DB	'I'	; INDIRECT
01F8 49		DB	IND-RTN	
01F9 54		DB	'T'	; TAPE LOADER
01FA 14		DB	TAPE-RTN	
01FB 4C		DB	'L'	; LOAD MEMORY POINTER
01FC 37		DB	LOD-RTN	
01FD 4A		DB	'J'	; JUMBO - LOAD 2 BYTES
01FE 2C		DB	DOUBLE-RTN	
01FF FF		DB	-1	

; RST1 IS THE INITIALIZATION ROUTINE
 ; IT SETS UP THE WORMHOLES USED IN THE
 ; MONITOR, THEN CHECKS FOR A SECOND ROM
 ; AND CALLS IT IF IT IS THERE

0200	RTN	EQU	\$	; ALL ROUTINES REFERENCE TO HERE
0200 11160C	RST1:	LXI	D, SRA4	
0203 21EA03		LXI	H, INITER	
0206 01EAF		LXI	B, INITLEN	
0209 CD0001		CALL	MOVE	; INIT WORMHOLES
020C 3A0004		LDA	400H	; GET 1ST BYTE OF 2ND ROM
020F 3C		INR	A	; IF IT WAS NOT 0FFH
0210 C40004		CNZ	400H	; CALL IT
0213 FB		EI		; TURN ON INTERRUPTS

; TAPE IS THE BOOTSTRAP LOADER ROUTINE
 ; IT EXPECTS TO GET A B, P, OR C OR IT WILL
 ; WAIT FOR ONE

0214 CD9203	TAPE:	CALL	CLEAR	
0217 CDA103	START:	CALL	LCFLD	; GET A CASE-FOLDED B OR P
021A 4F		MOV	C, A	; SAVE THE B OR P FOR ECHOING.
021B FE50		CPI	'P'	; POLYPHASE


```

021D CABA02      JZ      POLY
0220 D642        SUI      'B'      ;BYTE STANDARD
0222 CAC702      JZ      BITE
0225 3D          DCR      A
0226 CAF302      JZ      HEAD      ;CONTINUE
0229 C31702      JMP      START    ;NOT A B OR C OR P : TRY AGAIN

```

```

;
;
;DOUBLE GETS A HEX NUMBER FROM THE CONSOLE
;AND LOADS IT INTO THE NEXT 2 BYTES IN MEMORY
;(OR A REGISTER) LOW ORDER BYTE FIRST
;J<HEX #>(CR)
;

```

```

022C CDAA03      DOUBLE: CALL    HEXC
022F EB          XCHG
0230 2A500C      LHL      WINDOW
0233 73          MOV      M,E
0234 23          INX      H
0235 72          MOV      M,D
0236 C9          RET
;
;LOD LOADS WINDOW WITH THE NUMBER
;FOLLOWING THE L COMMAND
;L<HEX NUMBER>(CR)
;

```

```

0237 CDAA03      LOD:      CALL    HEXC
023A 22500C      SVW:      SHLD    WINDOW
023D C9          RET
;
;ADDIT MOVES THE MEMORY POINTER (WNDOW)
;BY FORWARD OR BACKWARDS BY
;THE VALUE IN E (D=0)
;E IS OFFSET BY 16
;

```

```

023E 2A500C      ADDIT:  LHL      WINDOW
0241 01F0FF      LXI      B,-16
0244 09          DAD      B
0245 19          DAD      D
0246 C33A02      JMP      SVW
;
;IND LOADS THE NEXT 2 BYTES IN MEMORY
;INTO WINDOW
;

```

```

0249 2A500C      IND:      LHL      WINDOW
024C 5E          MOV      E,M
024D 23          INX      H
024E 56          MOV      D,M
024F EB          XCHG
0250 C33A02      JMP      SVW
;
;STORE STORES THE BYTE IN L (ENTERED
;FROM CONSOLE) INTO MEMORY
;

```



```
0253 7D      STORE:  MOV      A,L
0254 2A500C  LHL      WINDOW
0257 77      MOV      M,A
0258 C9      RET
```

```
;
;X EXECUTES ONE INSTRUCTION POINTED TO
;BY SAVPC AND RETURNS TO RST7
```

```
;
X:  POP      H      ;2 DUMMY POPS
    POP      H      ;GET PC
    SHLD     SAVPC  ;SAVE IT
    POP      H
    POP      H      ;GET REGISTERS
    POP      D
    POP      B
    POP      PSW
    XTHL     ;RESTORE PC
    LHL      SAVPC
    XTHL
    EI
    OUT      12      ;ENABLE SINGLE STEP LOGIC
    RET      ;GO TO USER PGM.
```

```
;
;G ACTS THE SAME AS X BUT DOES NOT
;ENABLE SINGLE STEP LOGIC, AND
;THEREFORE DOES NOT RETURN
;TYPING CTL-Z WILL RETURN FROM THE
;PROGRAM BEING EXECUTED AND SAVE
;ALL REGISTERS
;IF A RST7 IS ENCOUNTERED THIS WILL
;ALSO HAPPEN
```

```
;
G:  POP      H
    POP      H
    SHLD     SAVPC
    POP      H
    POP      H
    POP      D
    POP      B
    POP      M
    XTHL
    LHL      SAVPC
    XTHL
    RET
```

```
;
;SETR POINTS TO ONE OF THE 8080 REGISTERS
;AS SAVED IN MEMORY
;MAY BE USED WITH JUMBO COMMAND TO
;SET A REGISTER PAIR, OR WITH OTHER
;COMMANDS TO SET INDIVIDUAL REGISTERS
;S<P/H/D/B/A>
```

```
;
027C CDA103  SETR:  CALL  LCFLD  ;GET REGISTER DESIGNATION
027F 218F02  LXI    H,RTAB ;AND LOOK UP POSITION
```



```

0282 CDD801      CALL    LOOKUP
0285 D8          RC      ;RETURN IF NOT VALID REGISTER
0286 210200      LXI     H,2
0289 54          MOV     D,H      ;SET D-0
028A 39          DAD     SP      ;ADD 2 TO STACK POINTER (FOR CAL
028B 19          DAD     D       ;ADD REG. POSITION
028C C33A02      JMP     SVW     ;PUT IN WINDOW

```

```

;
;RTAB IS A TABLE OF THE 8080 REGISTERS
;AND THEIR RELATIVE POSITIONS AS
;STORED ON THE STACK

```

```

028F 50      RTAB:  DB    'P'
0290 00      DB    0
0291 48      DB    'H'
0292 04      DB    4
0293 44      DB    'D'
0294 06      DB    6
0295 42      DB    'B'
0296 08      DB    8
0297 41      DB    'A'
0298 0A      DB    10
0299 FF      DB    -1

```

```

;
;GET IS USED BY THE LOADER TO GET
;C BYTES AND STORE THEM AT HL TO HL+C-1

```

```

029A AF      GET:   XRA     A
029B 47      MOV     B,A
029C CDA502  GET1:  CALL    TI
029F 77      MOV     M,A      ;PUT BYTE IN MEMORY
02A0 23      INX     H       ;INCREMENT POINTER
02A1 0D      DCR     C
02A2 C29C02  JNZ     GET1

```

```

;
;TI GETS A CHARACTER FROM WH2
;AND KEEPS A CHECKSUM IN B
;D IS USED AS A TEMPORARY

```

```

02A5 CD280C  TI:    CALL    WH2
02A8 57      MOV     D,A
02A9 80      ADD     B
02AA 47      MOV     B,A
02AB 7A      MOV     A,D
02AC C9      RET

```

```

;
;SETUP PUTS IMMEDIATE BYTES INTO BAUD RATE GEN.,
;AND THEN USART CTL PORT. THE TERM CHAR IS 00H,
;WHICH IS ALSO XMTED TO USART, LEAVING IT IDLING.
;WHEN DONE, JUMPS TO LOC. AFTER
;IMMED BYTES.

```

```

02AD E1      SETUP: POP     H      ;GET ADDR FOLLOWING CALL TO SETU
02AE 7E      MOV     A,M      ;GET DATA FOR BRG

```



```

02AF D304      OUT      4      ;PUT IN BRG
02B1 23      SET1:  INX      H      ;NEXT BYTE
02B2 7E      MOV      A,M
02B3 D301      OUT      1      ; USART PORT
02B5 B7      ORA      A      ;WAS THAT 00H?
02B6 C2B102    JNZ      SET1     ;NO,NEXT BYTE
02B9 E9      PCHL      ;JUMP TO THE 00H, EXEC. IT AS A
                                ;CONTINUE

```

;POLY AND BITE CONTAIN SETUP
 ;INFORMATION FOR POLYPHASE OR BYTE
 ;OPERATION OF THE USART

```

02BA CDAD02    POLY:  CALL  SETUP  ;THE NEXT BYTES GOTO BRG AND USART
02BD 05      * 25H IF 4.0 IN DB 005H ;TO BRG: SELECT DEV 0, 2400 BAUD
02BE AA      RAM DB 0AAH ;FAKE SYNCH CHAR IF USART EXPECTS IT
02BF 40      DB 040H ;INTERN. RESET. GETS USART TO MODE
02C0 0C      DB 00CH ;MODE CODE FOR SYNCHRONOUS, 8 BITS,
                                ;NO PARITY, INTERN. SYNC, 2 SYNC CHAR
02C1 E6      DB 0E6H ;FIRST SYNC CHAR. TO SRCH FOR
02C2 E6      DB 0E6H ;2ND SYNC CHAR.
02C3 00      DB 000H ;LEAVE COMMAND AT 00H (IDLE), RT
02C4 C3CF02    JMP  NAMED
02C7 CDAD02    BITE:  CALL  SETUP
02CA 06      * 26H IF 4.0 IN DB 006H ;TO BRG: SEL. DEV. 0, 300 BAUD
02CB AA      DB 0AAH ;FAKE SYNCH CHAR IF USART EXPECTS IT
02CC 40      DB 040H ;INT. RST. GETS US TO MODE LEVEL
02CD CE      DB 0CEH ;MODE: ASYNCH., 8 BITS, NO PARITY
                                ;2 STOP BITS, 16X CLOCK SCALING
02CE 00      DB 000H ;FOR NOW COMMAND IS IDLE
02CF 21540C    NAMED: LXI  H,FNAME
02D2 79      MOV  A,C ;ECHO THE B OR P
02D3 CD240C    CALL WHI
02D6 CD8D03    CALL CROUT ;OUTPUT CR
02D9 0E09      MVI  C,9

```

;NAM0 GETS THE NAME OF THE FILE
 ;TO BE LOADED FROM THE CONSOLE

```

02DB CD200C    NAM0:  CALL  WH0
02DE CD240C    CALL  WH1 ;ECHO CHAR.
02E1 FE0D      CPI  13
02E3 CAEC02    JZ   NAM ;DONE IF CR
02E6 77      MOV  M,A ;STORE IN MEMORY
02E7 23      INX  H
02E8 0D      DCR  C
02E9 C2DB02    JNZ  NAM0 ;DONE IF 9 CHARACTERS
02EC 3620      NAM:  MVI  M,20H ;FILL OUT WITH BLANKS
02EE 23      INX  H
02EF 0D      DCR  C
02F0 F2EC02    JP   NAM

```

;HEAD SEARCHES FOR A RECORD HEADER
 ;AND STORES IT AT RNAME

;COMP THEN COMPARES THE NAME AGAINST
 ;THE NAME IT IS SEARCHING FOR
 ;GETS NEXT HEADER IF NOT MATCH
 ;AFTER DISPLAYING NAME AND RECORD NUMBER
 ;IF CHECKSUM ERROR GOES TO ERROR

HEAD: LXI H,DNAME
 PUSH H ;ANYTHING RETURNING AFTER HERE
 ;WILL DISPLAY THE RECD NAME AND NO.
 HEAD6: MVI A,096H ;START USART READING, ENTER SEARCH
 ;IF SYNCHRONOUS, AND START MOTOR,
 ;TO USART CTL PORT
 OUT 1
 CALL WH2
 HEAD7: CPI 0E6H ;SYNC CHAR.
 JNZ HEAD6 ;RESYNC USART
 CALL WH2
 CPI 001H ;SOH CHAR.
 JNZ HEAD7
 LXI H,RNAME
 MVI C,14
 CALL GET ;GET HEADER
 JNZ ERROR
 LXI H,RNAME ;COMPARE NAMES
 LXI D,FNAME
 MVI C,8
 COMP: LDAX D
 CMP M
 RNZ ;NOT MATCH
 INX D
 INX H
 DCR C
 JNZ COMP ;FALL THRU IF MATCH
 LHL D ;GET LOAD ADDRESS AND
 LDA RLEN ;LENGTH IN PREPARATION FOR LOADING
 MOV C,A
 LDA RTYPE ;CHECK RECORD TYPE
 ORA A
 JZ GETD ;DATA, LOAD INTO RAM
 DCR A
 JZ COMNT ;COMMENT, DISPLAY IT
 DCR A
 JZ STOP ;END OF FILE, STOP TAPE
 DCR A
 RNZ ;NOT TYPES 0-3, TRY AGAIN
 OUT 1 ;STOP TAPE
 PCHL ;AUTO-EXECUTE, GO TO PGM.

;WERE DONE LOOKING AT OR READING A RECD, SO DISPLAY
 ;NAME AND RECD# OF LAST SEEN RECD.

DNAME: LXI B,RNAME
 MVI D,8
 DNAM2: STC
 CALL FLDSY

0342 015C0C
 0345 1608
 0347 37
 0348 CD7C01

49D4
 NA
 02F3 214203
 02F6 E5
 02F7 3E96

49DB
 02F9 D301
 02FB CD280C
 02FE FEE6
 0300 C2F702
 0303 CD280C
 0306 FE01
 0308 C2FE02
 030B 215C0C
 030E 0E0E
 0310 CD9A02
 0313 C27003
 0316 215C0C
 0319 11540C
 031C 0E08

49F9
 031E 1A
 031F BE
 0320 C0
 0321 13
 0322 23
 0323 0D
 0324 C21E03
 0327 2A670C
 032A 3A660C
 032D 4F
 032E 3A690C
 0331 B7
 0332 CA6C03
 0335 3D
 0336 CA5F03
 0339 3D
 033A CA7503
 033D 3D
 033E C0
 033F D301
 0341 E9

4A13.6
 18963.1
 4A17

C2D449

C2DB49

03 RNZ

C2F949

LXI H, 4A20

JMP GET

03 JZ

01 JZ

02 JZ

03 JZ

04 JZ

05 JZ

06 JZ

07 JZ

08 JZ

09 JZ

10 JZ

11 JZ

12 JZ

13 JZ

14 JZ


```

034B 15          DCR      D
034C C24703      JNZ      DNAM2
034F CD9703      CALL     BLK
0352 2A640C      LHL     RRN      ;DISPLAY R/N
0355 EB          XCHG
0356 CDD103      CALL     DEOUT
0359 CD8D03      CALL     CROUT
035C C3F302      JMP      HEAD
;
;COMNT DISPLAYS COMMENTS ON THE SCREEN
;
035F CD280C      COMNT:   CALL     WH2      ;ECHO TAPE ON CRT FOR A COMMENT
0362 CD7F00      CALL     DSPLY
0365 0D          DCR      C
0366 C25F03      JNZ      COMNT
0369 C3F702      JMP      HEAD6
;
;GETD GETS DATA AND RETURNS IF CS2 IS GOOD
;OTHERWISE IT STOPS THE TAPE
;AND PRINTS A "?"
;
036C CD9A02      GETD:    CALL     GET
036F C8          RZ
0370 3E3F      ERROR:   MVI      A,'?'
0372 CD7F00      CALL     DSPLY
0375 CD8D03      STOP:   CALL     CROUT
0378 AF          XRA      A
0379 D301      OUT      1      ;OUTPUT NUL TO STOP USART
037B E1          POP     H      ;CLEAN UP STACK
037C C31702      JMP      START
;
;***** UTILITY SUBROUTINES *****
;
;THESE SUBROUTINES MAY BE USED EXTERNALLY, SO WE WANT
;THEM IN KNOWN LOCATIONS.
;
;HEX08 OUTPUTS 8 BYTES FROM THE ADDRESS POINTED
;TO BY D,E LEAVING D,E POINTING TO THE NEXT
;LOCATION IN MEMORY. IT PUTS THE BYTES OUT
;IN HEX WITH A SPACE BETWEEN THEM AND A
;CARRAGE RETURN AT THE END OF THE LINE.
;
;D2 IS THE ACTUAL LOOP WHICH HEX08 USES
;IT PUTS OUT THE NUMBER OF BYTES IN B
;INCRMENTS D AND THEN PUTS OUT A
;CARRAGE RETURN.
;
FOR HEX016
037F 0608      HEX08:   MVI      B,8      16 - FOR HEX016
0381 CD9703      D2:     CALL     BLK
0384 1A          LDAX     D
0385 CDD603      CALL     BYTE
0388 13          INX      D
0389 05          DCR      B
038A C28103      JNZ      D2
  
```



```

;
;CROUT, CLEAR, BLK, AND TABBER
;OUTPUT A CARRIAGE RETURN, FORM FEED, BLANK,
;OR HORIZONTAL TAB TO THE CRT DRIVER, RESPECTIVELY
;
038D 3E0D    CROUT: MVI    A,CR    ;PUT CAR RETRN ON CONSOLE DSPLY
038F C37F00      JMP    DSPLY
0392 3E0C    CLEAR: MVI    A,12    ;CTL-L (FORM FEED)
0394 C37F00      JMP    DSPLY
0397 3E20    BLK:   MVI    A,' '    ;SPACE
0399 C37F00      JMP    DSPLY
039C 3E09    TABBER: MVI    A,9     ;CTL-I TAB
039E C37F00      JMP    DSPLY
;
;LCFLD (LOWER CASE FOLD). GETS A CHAR FROM WH0.
;IF AN LC CHAR WAS FOLDED, CY=0,ELSE CY=1.
;WATCH RUBOUTS! THEY'RE FOLDED TO 5FH FROM NORMAL 7FH.
;
03A1 CD200C    LCFLD: CALL    WH0    ;GET CHAR
03A4 FE60      CPI    060H    ;IF UPPER CASE,FORGET IT
03A6 D8        RC            ;UC LETTERS ARE LESS THAN 60H
03A7 D620      SUI    020H    ;FOLD THIS LOWER CASE LETTER ON
03A9 C9        RET
;
;HEXC INPUTS VARIABLE LENGTH HEX FROM WH0, ECHOING ON DS
;TERM CHAR IS ANY NON-HEX CHAR, AND IT IS RETURNED IN B.
;DIGIT COUNT RETURNED IN C.  VALUE RETURNED IN H,L
;
03AA 210000    HEXC:  LXI    H,0     ;ZERO CONVERSION BUFFER
03AD 4D        MOV    C,L
03AE CDA103    NXNYB: CALL    LCFLD  ;GET A CASE-FOLDED CHAR FROM WH0
03B1 47        MOV    B,A    ;SAVE THIS CHAR SO IT CAN BE USE
;IT WAS THE TERM CHAR.
03B2 FE30      CPI    '0'    ;RETURN IF LESS THAN ASCII 0
03B4 D8        RC
03B5 CD240C    CALL    WH1    ;ECHO EACH CHAR. VALID OR NOT
03B8 D630      SUI    '0'    ;CHANGE ASCII INTO BINARY 0-15
03BA FE0A      CPI    10
03BC DAC703    JC     NXNB1
03BF D607      SUI    7
03C1 FE0A      CPI    10
03C3 D8        RC            ;RETURN IF NOT HEX
03C4 FE10      CPI    16
03C6 D0        RNC            ;RETURN IF NOT HEX
03C7 0C        NXNB1: INR    C     ;COUNT # OF HEX CHARACTERS
03C8 29        DAD    H        ;SHIFT HL
03C9 29        DAD    H        ;OVER
03CA 29        DAD    H        ;FOR NEXT
03CB 29        DAD    H        ;DIGIT
03CC B5        ORA    L        ;OR IN NEW DIGIT
03CD 6F        MOV    L,A
03CE C3AE03    JMP    NXNYB
;
;DEOUT OUTPUTS DE TO THE SCREEN

```



```

;AS 4 HEX DIGITS
;
03D1 7A      DEOUT:  MOV     A,D
03D2 CDD603      CALL    BYTE
03D5 7B      MOV     A,E
;
;BYTE OUTPUTS THE ACCUMULATOR
;AS 2 HEX DIGITS TO THE SCREEN
;
03D6 F5      BYTE:   PUSH    PSW
03D7 0F      RRC
03D8 0F      RRC
03D9 0F      RRC
03DA 0F      RRC
03DB CDDF03    CALL    HEXO
03DE F1      POP     PSW
;
;HEXO OUTPUTS 1 HEX DIGIT TO
;THE SCREEN - THE UPPER HALF
;OF A IS MASKED WITH ZEROS
;
03DF E60F      HEXO:   ANI     15
03E1 C690      ADI     90H
03E3 27      DAA
03E4 CE40      ACI     40H
03E6 27      DAA
03E7 C37F00    JMP     DSPLY ;OUTPUT HEX DIGIT AND USE RETURN
;
;***** INITIALIZATION PARAMETERS *****
;THE FOLLOWING INFORMATION IS USED ON FPRST OR POC
;TO SETUP THE STARTING SYSTEM CONTEXT.
;
1000          STACK   EQU     01000H ; USED IN AN LXI,SP
;
;THE FOLLOWING BLOCK IS COPIED DIRECTLY OVER
;SYSTEM RAM STARTING AT SRA4.
;
03EA 5400      INITER:  DW      USRTSR ;VI3 USART INTERRUPT
03EC 0D01      DW      KSR          ;VI2: THE STANDARD KBD INT
03EE 6400      DW      IORET        ;WAKEUP: NOTHING FOR NOW
03F0 1701      DW      SAVE         ;VI0: SINGLE STEP INT GOES BACK
03F2 FC        DB      0FCH        ;VIDEO SCRN ENDS AT FC00H-1
03F3 F8        DB      0F8H        ;VIDEO SCRN HOME AT F800H
03F4 CD6A00    CALL    KI           ;WORMHOLE 0: INIT TO KBD AT F8H
03F7 C9        RET                ;STANDARD PART OF ANY WORMHOLE
03F8 CD7F00    CALL    DSPLY        ;WORMHOLE 1: INIT TO VIDEO DSPL
03FB C9        RET
03FC CD7800    CALL    USRTI        ;WORMHOLE 2: INIT TO USART AT 0
03FF C9        RET
0000          END

```


SYMBOLS SORTED BY NAME

ADDIT 023E	BITE 02C7	BLK 0397	BYTE 03D6	CLEAR 0392
CLINE 00D0	CLOCK 0030	COMD 01BA	COMNT 035F	COMP 031E
CR 000D	CROUT 038D	CRR 00E6	CTLX 0018	CTLZ 001A
CURP 00DC	D2 0381	DEOUT 03D1	DISP 0134	DISP1 0148
DNAM2 0347	DNAME 0342	DOUBL 022C	DSPLY 007F	ERROR 0370
FF 000C	FF1 00F2	FFR 00EF	FLAGS 0187	FLDSY 017C
FNAME 0C54	G 026C	GET 029A	GET1 029C	GETD 036C
HEAD 02F3	HEAD6 02F7	HEAD7 02FE	HEXC 03AA	HEXO 03DF
HEX08 037F	IND 0249	INITE 03EA	INITL FFEA	IOPUT 0060
IORET 0064	KBUFF 0C0C	KI 006A	KI1 006E	KSR 010D
LCFLD 03A1	LOD 0237	LOOKU 01DB	MMOD 0198	MMOD1 01A8
MOVE 0100	MSG 016D	MTBL 01E8	NAM 02EC	NAM0 02DB
NAMER 02CF	NORM 00B4	NXNB1 03C7	NXNYB 03AE	POLY 02BA
POS 0C0E	POS2 0C4A	POS3 0C4E	RADR 0C67	RB 007F
RBR 00E4	RBUFF 0C0A	RESET 0000	RLEN 0C66	RNAME 0C5C
RRN 0C64	RST1 0200	RTAB 028F	RTN 0200	RTYPE 0C69
SAVE 0117	SAVPC 0C52	SCEND 0C1E	SCND2 0C48	SCND3 0C4C
SCRH2 0C49	SCRH3 0C4D	SCRHM 0C1F	SCRL 00B9	SET1 02B1
SETR 027C	SETUP 02AD	SRA1 0C10	SRA2 0C12	SRA3 0C14
SRA4 0C16	SRA5 0C18	SRA7 0C1C	SS 0038	STACK 1000
START 0217	STOP 0375	STORE 0253	SVW 023A	TAB 0009
TABBE 039C	TANI 0C04	TAPE 0214	TBUFF 0C08	TI 02A5
TIME 0040	TIME2 0047	TIMER 0C00	USRTI 0078	USRTS 0054
VCB2 0C48	VCB3 0C4C	VI2 0028	VI3 0020	VI4 0018
VI5 0010	VI6 0008	VT 000B	VTR 00F9	WAKEU 0C1A
WARM 0121	WH0 0C20	WH1 0C24	WH2 0C28	WH3 0C2C
WH4 0C30	WH5 0C34	WH6 0C38	WH7 0C3C	WH8 0C40
WH9 0C44	WINDO 0C50	WIPE 00D5	X 0259	

SYMBOLS SORTED BY VALUE

RESET 0000	VI6 0008	TAB 0009	VT 000B	FF 000C
CR 000D	VI5 0010	CTLX 0018	VI4 0018	CTLZ 001A
VI3 0020	VI2 0028	CLOCK 0030	SS 0038	TIME 0040
TIME2 0047	USRTS 0054	IOPUT 0060	IORET 0064	KI 006A
KI1 006E	USRTI 0078	DSPLY 007F	RB 007F	NORM 00B4
SCRL 00B9	CLINE 00D0	WIPE 00D5	CURP 00DC	RBR 00E4
CRR 00E6	FFR 00EF	FF1 00F2	VTR 00F9	MOVE 0100
KSR 010D	SAVE 0117	WARM 0121	DISP 0134	DISP1 0148
MSG 016D	FLDSY 017C	FLAGS 0187	MMOD 0198	MMOD1 01A8
COMD 01BA	LOOKU 01DB	MTBL 01E8	RST1 0200	RTN 0200
TAPE 0214	START 0217	DOUBL 022C	LOD 0237	SVW 023A
ADDIT 023E	IND 0249	STORE 0253	X 0259	G 026C
SETR 027C	RTAB 028F	GET 029A	GET1 029C	TI 02A5
SETUP 02AD	SET1 02B1	POLY 02BA	BITE 02C7	NAMER 02CF
NAM0 02DB	NAM 02EC	HEAD 02F3	HEAD6 02F7	HEAD7 02FE

COMP 031E	DNAME 0342	DNAM2 0347	COMNT 035F	GETD 036C
ERROR 0370	STOP 0375	HEX08 037F	D2 0381	CROUT 038D
CLEAR 0392	BLK 0397	TABBE 039C	LCFLD 03A1	HEXC 03AA
NXNYB 03AE	NXNB1 03C7	DEOUT 03D1	BYTE 03D6	HEX0 03DF
INITE 03EA	TIMER 0C00	TANI 0C04	TBUFF 0C08	RBUFF 0C0A
KBUFF 0C0C	POS 0C0E	SRA1 0C10	SRA2 0C12	SRA3 0C14
SRA4 0C16	SRA5 0C18	WAKEU 0C1A	SRA7 0C1C	SCEND 0C1E
SCRHM 0C1F	WH0 0C20	WH1 0C24	WH2 0C28	WH3 0C2C
WH4 0C30	WH5 0C34	WH6 0C38	WH7 0C3C	WH8 0C40
WH9 0C44	SCND2 0C48	VCB2 0C48	SCRH2 0C49	POS2 0C4A
SCND3 0C4C	VCB3 0C4C	SCRH3 0C4D	POS3 0C4E	WINDO 0C50
SAVPC 0C52	FNAME 0C54	RNAME 0C5C	RRN 0C64	RLEN 0C66
RADR 0C67	RTYPE 0C69	STACK 1000	INITL FFEA	

THATS ALL, FOLKS!

COMP 031E	NAME 0342	NAME 0347	COMNT 035F	GETD 036C
ERROR 0370	STOP 0375	HEX08 037F	D3 0381	CROUT 038D
CLEAR 0392	BLK 0397	TAB08 039C	LCFLD 03A1	HEXC 03AA
INXN08 03AE	INXN01 03C7	DEOUT 03D1	BYTE 03DE	HEX0 03DE
INITE 03EA	TIMER 0C08	TANT 0C04	TBUF0 0C08	RBUF0 0C0A
KBUF0 0C0C	POS 0C0E	GRA1 0C10	GRA2 0C12	GRA3 0C14
GRA4 0C16	GRA5 0C18	WAK0U 0C1A	GRA7 0C1C	SEEND 0C1E
SCRNM 0C1F	WH0 0C20	WH1 0C24	WH2 0C28	WH3 0C2C
WH4 0C30	WH5 0C34	WH6 0C38	WH7 0C3C	WH8 0C40
WH9 0C44	SCND2 0C48	VCB2 0C48	SCRN2 0C49	POS2 0C4A
SCND3 0C4C	VCB3 0C4C	SCRN3 0C4D	POS3 0C4E	WIND0 0C50
SAVFC 0C52	FNAME 0C54	RNAME 0C5C	RNM 0C64	RTEN 0C68
RADR 0C6F	RTYPE 0C69	STACK 1000	INITF 1F0A	

THATS ALL, FOLKS!

Appendix E Onboard RAM dumper for the 4.0 monitor

E3

; ***** ONBOARD DUMPER FOR 4.0 *****

; THIS IS A POLYFORMAT DUMPER FOR ABSOLUTE
; DATA WHICH RUNS FROM C6A TO D9F (OR SO), START ADDRESS
; C6AH. WHEN RUN, IT ACTS LIKE 4.0 MONITOR TAPE LOAD IN
; THE WAY IT ACCEPTS ENCODING SPECIFICATION (B OR P) AND
; FILE NAME. THEN IT EXPECTS TWO HEX NUMBERS FOR
; START AND END DUMP ADDRESSES. EACH RECORD DUMPED SHOWS
; ADDRESS USED IN HEX ON SCREEN. WHEN DONE, IT PUTS OUT
; AN "END" TYPE RECORD AND CLEARS SCREEN, READY
; FOR ANOTHER DUMP.

; ORIGINAL 2.2 DUMPER SYSTEM WRITTEN BY DAVID FAIMAN
; REWRITTEN, DOCUMENTED AND CONVERTED TO ONBOARD FOR 4.0
; BY R.L.DERAN

0C20	WH0	EQU	0C20H
0C24	WH1	EQU	0C24H
0C16	SRA4	EQU	0C16H
02AD	SETUP	EQU	02ADH
03AA	HEXC	EQU	03AAH
03D1	DEOUT	EQU	03D1H
0C5A		ORG	0C5CH-2
0C5A	LENGTH:	DS	2 - LENGTH
0C5C	WNAME:	DS	8
0C64	WRN:	DS	2
0C66	WLEN:	DS	1
0C67	WADR:	DS	2 - START ADDR
0C69	WTYPE:	DS	1 - RECORD TYPE
0C6A 21450D	START:	LXI	H, TISR
0C6D 22160C		SHLD	SRA4
0C70 3E0C	STAR2:	MVI	A, 0CH ; FORM FEED
0C72 CD240C		CALL	WH1 ; CLEAR SCREEN
0C75 CD200C		CALL	WH0
0C78 CD240C		CALL	WH1
0C7B FE42		CPI	'B'
0C7D CA920C		JZ	BITE
0C80 FE50		CPI	'P'
0C82 C2700C		JNZ	STAR2
0C85 CDAD02	POLY:	CALL	SETUP
0C88 05	25H if 4.0 in RAM	DB	005H - BAUD RATE
0C89 AA		DB	0AAH - FAKE SYNC CHAR
0C8A 40		DB	040H - INTERNAL RESET
0C8B 0C		DB	00CH - MODE CODE - SYNCHRONOUS
0C8C E6		DB	0E6H - FIRST SYNC CHAR - 8 BIT
0C8D E6		DB	0E6H - 2ND SYNC CHAR TO SRCH FOR NO PARITY
0C8E 00		DB	000H - END INTERNAL SYNC
0C8F C39A0C		JMP	NAMER
0C92 CDAD02	BITE:	CALL	SETUP
0C95 06	26H if 4.0 in RAM	DB	006H - BAUD RATE
0C96 AA		DB	0AAH - FAKE SYNC CHAR
0C97 40		DB	040H - INTERNAL RESET

/CHANGES FOR STRING

CALL CROUT INSTEAD

SYNCHRONOUS
8 BIT
NO PARITY
INTERNAL SYNC
2 STOP BITS?

SEE PG. 8
OF SERIAL
PORT SECTION
OF MANUAL
FOLLOWS FIG. 4
OF PG. 7


```

0C98 CE      DB      0CEH -MODE- ASYNCH, 8BITS, NO PARITY, 2 STOP BITS, 16X CLOCK
0C99 00      DB      000H

```

```

;
; NAMEING ROUTINE
;

```

```

0C9A 210000  NAMER:  LXI      H,0
0C9D 22640C      SHLD     WRN
0CA0 0E08      MVI      C,8      ;BLANK NAME FIELD
0CA2 21630C      LXI      H,WNAME+7
0CA5 3620      NAM:     MVI      M,020H
0CA7 2B      DCX      H      ;BACKUP H TO WNAME
0CA8 0D      DCR      C
0CA9 C2A50C      JNZ      NAM
0CAC 23      INX      H
0CAD 0E08      MVI      C,8
0CAF CD180D      CALL     CRLF
0CB2 CD200C      NAM0:    CALL     WH0
0CB5 CD240C      CALL     WH1
0CB8 FE0D      CPI      00DH      ;CR
0CBA CAC30C      JZ       DUMPC
0CBD 77      MOV      M,A
0CBE 23      INX      H
0CBF 0D      DCR      C
0CC0 C2B20C      JNZ      NAM0
0CC3 AF      DUMPC:    XRA      A      ; CHANGE A TO 0
0CC4 32690C      STA      WTYPE
0CC7 CD180D      CALL     CRLF
0CCA CDAA03      SIZE:    CALL     HEXC
0CCD 22670C      SHLD     WADR
0CD0 78      MOV      A,B
0CD1 CD240C      CALL     WH1
0CD4 EB      XCHG
0CD5 CDAA03      CALL     HEXC
0CD8 CD180D      CALL     CRLF
0CDB 7D      MOV      A,L
0CDC 93      SUB      E
0CDD 6F      MOV      L,A
0CDE 7C      MOV      A,H
0CDF 9A      SBB      D
0CE0 67      MOV      H,A
0CE1 225A0C      SHLD     LENGTH
0CE4 CDF60C      CALL     DUMPR
0CE7 3E02      ENDC:    MVI      A,2
0CE9 32690C      STA      WTYPE
0CEC 3D      DCR      A
0CED 32660C      STA      WLEN
0CF0 CD540D      CALL     DUMP
0CF3 C36A0C      JMP      START

```

```

;
; DUMP DATA RECORDS
;

```

```

0CF6 215B0C  DUMPR:  LXI      H,LENGTH+1
0CF9 7E      MOV      A,M
0CFA B7      ORA      A
0CFB CA100D      JZ       OVER
0CFE 35      DCR      M
0CFF AF      XRA      A
0D00 32660C      STA      WLEN

```

9/10 } FORMAT
 NAME CR } NAME
 0.4 } TYPE

/ (LXI H, TYPE) TYPE?
 CALL STRG

CALL WH0
 CALL WH1
 SUB 30H
 STA WTYPE

/ PUSH PSW

DE, HL
 L H

POP PSW
 CPI 2 (end)
 RNZ --

eliminate for
 comment, absolute, data
 files. Use only
 for auto execute or
 end.
 RET instead of JMP START


```

0D03 CD540D      CALL    DUMP
0D06 2A670C      LHL    WADR
0D09 24          INR     H
0D0A 22670C      SHLD   WADR
0D0D C3F60C      JMP     DUMPR
0D10 2B          OVER:  DCX     H
0D11 7E          MOV     A,M
0D12 32660C      STA     WLEN
0D15 C3540D      JMP     DUMP

```

```

0D18 3E0D      ;
0D1A CD240C    CRLF:  MVI     A,0DH
0D1D C9        CALL    WH1
                     RET

```

```

;
; ROUTINE TO OUTPUT A RECORD
;

```

```

0D1E 0600      PUT:   MVI     B,0      ;CLEAR CHECKSUM
0D20 4F        MOV     C,A      ;PUT LENGTH OF RECORD IN C
0D21 7E        PUT0:  MOV     A,M
0D22 23        INX     H
0D23 F5        PUSH    PSW
0D24 80        ADD     B
0D25 47        MOV     B,A
0D26 F1        POP     PSW
0D27 CD340D    CALL    TO
0D2A 0D        DCR     C
0D2B C2210D    JNZ     PUT0
0D2E 78        MOV     A,B
0D2F 2F        CMA
0D30 3C        INR     A
0D31 C3340D    JMP     TO

```

```

;
; TAPE OUTPUT ROUTINE
;

```

```

0C08      TBUF:  EQU     0C08H
0D34 E5      TO:   PUSH    H
0D35 21080C  LXI     H,TBUF
0D38 F5      PUSH    PSW
0D39 7E      TO1:  MOV     A,M
0D3A B7      ORA     A
0D3B C2390D  JNZ     TO1
0D3E 23      INX     H
0D3F F1      POP     PSW
0D40 77      MOV     M,A
0D41 2B      DCX     H
0D42 34      INR     M
0D43 E1      POP     H
0D44 C9      RET

```

```

;
; TISR IS A SIMPLE USART READER WHICH WILL
; RE-TRANSMIT THE CHARACTER IN TBUF IF IT HAS NOT
; BEEN REPLACED BY THE WORMHOLE ROUTINE. IT
; DOES NOT CHECK THE FLAG, BECAUSE IT ASSUMES
; THAT THE PROGRAM CALLING THE WORMHOLE IS FASTER
; THAN THE USART AND SO IT ALWAYS HAS A VALID
; CHARACTER FOR US TO TAKE.
;

```

```

0D45 AF      TISR:  XRA     A

```

ADDIT NEED

WHY?


```

0D46 32080C      STA      TBUF
0D49 3A090C      LDA      TBUF+1
0D4C D300        OUT      0
0D4E E1          IORET:   POP      H
0D4F D1          POP      D
0D50 C1          POP      B
0D51 F1          POP      PSW
0D52 FB          EI
0D53 C9          RET

;
;
; DUMP PUTS OUT ONE COMPLETE RECORD.
; IT TURNS ON USART AND MOTORS, WAITS A WHILE
; FOR AN IRG, PUTS OUT 64 SYNCH CHARACTERS,
; DUMPS A RECORD ACCORDING TO THE WRITE CONTROL
; BLOCK AT WNAME (IT ALSO PUTS THE WCB
; ON THE RECORD AS HEADER), INCREMENTS THE RECORD
; NUMBER, STOPS USART AND MOTORS, AND RETURNS.
;
0D54 3E21      DUMP:     MVI      A, 021H
0D56 D301      OUT      1
0D58 2A670C    LHL      WADR
0D5B EB        XCHG
0D5C CDD103    CALL     DEOUT      ; DISPLAY THE ADDRESS WE'RE DUMPING
0D5F CD180D    CALL     CRLF
0D62 21FF8F    LXI      H, 08FFFH
0D65 2B        DELAY:    DCX      H
0D66 7C        MOV      A, H
0D67 B7        ORA      A
0D68 C2650D    JNZ      DELAY
0D6B 0E40      MVI      C, 64
0D6D 3EE6      MVI      A, 0E6H   ; SYNC CHARACTER
0D6F CD340D    DUMP0:    CALL     TO
0D72 0D        DCR      C
0D73 C26F0D    JNZ      DUMP0
0D76 3E01      MVI      A, 001H   ; START OF HEADER
0D78 CD340D    CALL     TO

;
; DUMP HEADER AND DATA RECORDS
;
0D7B 3E0E      MVI      A, 00EH   ; LENGTH OF HEADER RECORD
0D7D 215C0C    LXI      H, WNAME
0D80 CD1E0D    CALL     PUT
0D83 3A660C    LDA      WLEN
0D86 2A670C    LHL      WADR
0D89 CD1E0D    CALL     PUT
0D8C 21640C    LXI      H, WRN
0D8F 34        INR      M
0D90 AF        OFF:     XRA      A
0D91 CD340D    CALL     TO
0D94 CD340D    CALL     TO
0D97 CD340D    CALL     TO
0D9A D301      OUT      1
0D9C C9        RET
0000          END

; THESE PUSH OUT LAST BYTES FROM
; THE USART AND WH BUFFER PIPELIN
; TURN OFF MOTOR AND TRANSMITTER

```

DON'T NEED